Information Systems Frontiers
https://doi.org/10.1007/s10796-022-10267-4

Trends and Trajectories in the Software Industry: implications for the future of work

Samuli Laato[1,2] · Matti Mäntymäki[1] · A. K.M. Najmul Islam[1,3] · Sami Hyrynsalmi[3] · Teemu Birkstedt[1]

Accepted: 4 March 2022
© The Author(s) 2022

Abstract
In this study, we explore prominent contemporary technology trajectories in the software industry and how they are expected to influence the work in the software industry. Consequently, we build on cultural lag theory to analyze how technological changes affect work in software development. We present the results from a series of expert interviews that were analyzed using the Gioia method. Moreover, we identify a set of technology trends pertinent to software development from which we derive four main changes affecting the future of work in software development: (1) a shift toward scalable solutions, (2) increased emphasis on data, (3) convergence of IT and non-IT industries, and (4) the cloud as the dominant computing paradigm. Accordingly, this study contains insights into how technology (as an element of material culture) influences non-material culture, as exemplified by the work involved in software development.

Keywords Software development · Software industry · Digital transformation · Future of work · Changing nature of work · Cultural lag · Cultural lag theory

1 Introduction

Software development is undergoing a transformative change, both as an industry and as a profession (Bianchi et al., 2020; Koutsikouri et al., 2020; Maruping & Matook, 2020). Furthermore, tools and practices that improve automation, versatility, and scalability have become prominent (Schneckenberg et al., 2021), including continuous integration/continuous development (CI/CD) (Nogueira et al., 2018; Zhao et al., 2017) and cloud platforms managing the provision of hardware resources and the lower levels of the software stack.

However, it takes time for companies to react to available technologies and adjust their software development culture and practices (Ogburn, 1957; Schneckenberg et al., 2021; Suominen et al., 2014). From the perspective of practitioners, this implies that to evaluate the challenges and opportunities related to technological innovations (and make use of them), it is paramount to understand contemporary technology trends and developments (AL-Zahrani & Fakieh, 2020; Maruping & Matook, 2020; Wu, 2019). Moreover, as technology is continually advancing, research on pertinent contemporary technology trends should be constantly updated (Wong et al., 2021). Identifying these trends will help bridge the gap between research and practice (Gurcan and Kose, 2017; Gurcan and Cagiltay, 2019) and provide insights into the future of work in the software industry. Consequently, we address the following research questions (RQs) in this paper:

RQ1 *What are the most prominent contemporary technology trajectories in the software industry?*

RQ2 *How are they expected to influence work in software development?*

To answer the RQs, we conducted 18 expert interviews with seasoned professionals in leading positions within the software industry or academia who are actively dealing with the latest technologies in their profession. Moreover, we employ cultural lag theory (Ogburn, 1957) to describe and provide insights into the transformation processes through

✉ Samuli Laato
sadala@utu.fi

1 University of Turku, Turku, Finland

2 Tampere University, Tampere, Finland

3 LUT School of Engineering, LUT University, Lahti, Finland

Published online: 20 April 2022

which new technologies (material culture) influence the nonmaterial culture of software development. In so doing, we respond to the call for research on the transformation of software development (AL-Zahrani & Fakieh, 2020) in three ways. First, we identified 14 technology trends pertinent to contemporary software development. Second, we elucidated the transformation processes through which these changes could affect the nonmaterial culture of software development by applying the Gioia method (Gioia et al., 2013). Third, we theorized four aggregate dimensions of non-material cultural trends. This allowed us to discuss the implications of ongoing and future changes in the nonmaterial culture connected to the software industry on the changing nature of the work conducted by software developers.

Our paper contributes to the previous literature on contemporary technology trends and their impact on employment (Maruping & Matook, 2020) by exploring further prominent technologies discussed in previous IS literature. These include, for example, AI technologies (Collins et al., 2021; Bankins et al., 2022), DevOps (AL-Zahrani & Fakieh, 2020; Guşeilă et al., 2019), transition to remote work (Hafermalz, 2021; Hardill & Green, 2003; Waizenegger et al., 2020; Zamani & Pouloudi, 2021), the metaverse (Xi et al., 2022), augmented reality and robotics (Wang et al., 2021) and cloud computing (Schneckenberg et al., 2021). In addition, our work provides insights into labor market disruptions and the future of work (Drahokoupil & Fabo, 2016; Frey & Osborne, 2017; Healy et al., 2017) in the software industry.

The remainder of this study is structured as follows. First, we examine the extant literature on the changing nature of work within the software industry, followed by the introduction of our theoretical lens: cultural lag theory (Ogburn, 1957). Thereafter, we present the materials and methods for our empirical study, followed by the results. We conclude by discussing the key findings and implications of our results, limitations of the study, and opportunities for future research.

2 Background

2.1 The changing nature of work within the software industry

According to a report by the World Bank (2018), changes, transformations, and even disruptions that are driven by technology can be the main drivers of the changing nature of work. Digital transformation (DT) is a key area of IS research that addresses such changes (Reis et al., 2018). The main body of DT research has involved examining a wide range of phenomena related to shifts, mutations, and realignments driven by digital technology (Reis et al., 2018; Vial, 2019; Zhu et al., 2021). Importantly, the ramifications of DT transcend from societal to organizational and, ultimately, individual levels. Vial (2019) defines DT as *"a process that aims to improve an entity by triggering significant changes to its properties through combinations of information, computing, communication, and connectivity technologies.''* In addition to the general upswing of IS research on DT (e.g., Verhoef et al., 2021; Vial, 2019), some recent studies have examined DT in the context of the software industry (AL-Zahrani & Fakieh, 2020; Guşeilă et al., 2019; Klünder et al., 2019).

Software can be considered one of the main drivers of DT, meaning that any changes in the software industry are likely to cascade over to other industries through DT processes (Akter et al., 2020). Accordingly, software and software development are both driving and being affected by DT. There are many contemporary technologies that currently have (and are predicted to have) transformative effects on businesses. These include machine learning (ML) and deep learning (Brock & Von Wangenheim, 2019, Collins et al., 2021; Dwivedi et al., 2019; Magistretti et al., 2019; Laato et al., 2020; Wong et al., 2021), blockchain (Islam et al., 2019), and technological services such as cloud computing (Akter et al., 2020; Al-Ruithe et al., 2018; Wong et al., 2021). Moreover, these technologies are being further developed and shaped to fit the needs of specific industries (Frick et al., 2021). When software and technologies are employed in this manner, they trigger and direct DT processes (Duan et al., 2019; Hess et al., 2016; Matt et al., 2015; Vial, 2019). Accordingly, by identifying any underlying technological megatrends and opportunities they offer businesses, we can forecast upcoming implications beyond the value network of individual companies (Pappas et al., 2018; Verhoef et al., 2021; Vial, 2019).

Apart from the DT perspective, researchers have also examined recent trends within the software engineering industry. For example, through analyzing posted job advertisements (Gurcan & Cagiltay, 2019; Gurcan & Kose, 2017) and by focusing on how new software development paradigms have changed the composition of software and how it is developed (Hemon-Hildgen et al., 2020; Wiedemann et al., 2020). Overall, this body of literature is dealing with the same technology trends as the DT literature, although the importance of identifying trajectories is also highlighted due to the constant evolution of technology (Gurcan & Cagiltay, 2019; Gurcan & Kose, 2017; Wiedemann et al., 2020). This suggests that the ecosystem in which software is created and orchestrated is in a constant state of flux, which explains why recent research has argued that IS scholars should focus on emerging trends and trajectories in this field (Maruping & Matook, 2020; Estevam et al., 2020).

2.2 Cultural lag theory

We employ cultural lag theory (Ogburn, 1957) as our theorizing device because it allows us to explore current trends in material culture and predict upcoming changes to nonmaterial culture (Brinkman & Brinkman, 1997; Ogburn, 1957). Accordingly, it is particularly suitable for solving our research goals and for helping to make sense of empirical data on technology trajectories and their influence on the future of work.

Cultural lag theory is a macro-level theory that focuses on examining the sociocultural implications of developments in material culture (such as technology). Hence, the theory distinguishes between material and nonmaterial cultures. Material culture comprises physical objects, such as technologies, products, and services (Suominen et al., 2014). Accordingly, it can be understood that material culture also includes intangible digital technologies (i.e., all software) (Bertani et al., 2021). In contrast, nonmaterial culture relates to ideas, thoughts, beliefs, and ideologies. The central postulate of cultural lag theory is that material culture evolves more rapidly than nonmaterial culture, and nonmaterial culture adjusts to any changes imposed by material culture over time (Marshall, 1999; Ogburn, 1957; Ogburn, 1966). This process of nonmaterial culture adjusting to changes in material culture is called cultural lag. With respect to DT and changes within the software industry, cultural lag can be attributed to many factors, such as the sluggishness involved when companies have to hire a new workforce, retrain their employees, and change their working practices (cf. Marshall, 1999; Ogburn, 1957; Suominen et al., 2014).

Cultural lag theory focuses on changes that begin with new developments in material culture, which then cascade and propagate over to nonmaterial culture (Ogburn, 1957; Ogburn, 1966). Due to its focus on the influence of material culture on nonmaterial culture, cultural lag theory bears some resemblance to technological determinism (Brinkman & Brinkman, 2006). Technological determinism implies that certain technological advances are inevitable, arise organically (independent of the surrounding nonmaterial culture), and shape human culture in a deterministic way (Bimber, 1990). According to Bimber (1990), only those approaches that make the ontological claim of the deterministic outcomes of technology should adopt the label of technological determinism. Building on this argument, cultural lag theory does not imply deterministic outcomes; rather, it merely implies that the rate of change that new technologies impose on society is gradual and not instantaneous (Ogburn, 1957). Dafoe (2015) suggests that the term technological determinism should be toned down to refer to the *"autonomous and social-shaping tendencies of technology"* instead of all aspects of technology. It is equally important to make this distinction in cultural lag theory, as not all forms of technology have outcomes on nonmaterial culture. Accordingly, cultural lag theory is useful for understanding the implications of technological developments on the macro-level, instead of specific instances of technology adoption. Finally, technology is not the only aspect of material culture that transforms nonmaterial culture. For example, visual art, music, urban design, and architecture influence human interactions and, consequently, nonmaterial culture (Ogburn, 1957).

3 Methodology

To answer the research questions, we conducted an expert interview study (Meuser & Nagel, 2009). Our primary goal for the interviews, as described in RQ1, was to harness the expertise of employees in leading positions within the software industry and academia to share their thoughts on pertinent technology trends. Furthermore, as indicated by RQ2, we sought to identify how these trends influence the nonmaterial culture within the software industry. We adopted this broad view to align with the macro-level perspective espoused by cultural lag theory (Ogburn, 1957), although it should be noted that this approach involves certain boundary conditions pertaining to who we recruited for the interviews. Next, we discuss participant sampling in greater detail, followed by the interview process and subsequent data analysis.

3.1 Data collection

The data for the empirical section of this work were collected through thematic interviews (e.g., Gubrium & Holstein, 2001). To interview experts who had sufficient knowledge to provide insights into the research question, we established three guiding criteria for participant sampling. First, the informant was required to have worked in a prominent and unique position, either within the software industry or in an academic position, for the past five years. Second, the role of the work had to be focused in some way on software development. Third, to ensure comprehensive and rich data, we recruited informants without significant overlap in terms of their role, primary competence, and background. Keeping these criteria in mind, we followed the snowball sampling technique to find and recruit experts for the interviews. The process started with all authors suggesting names, discussing potential candidates, and contacting informants for the interviews. We particularly searched for respondents from Finland, which is a country with a high-technology industry. This also ensured the authors had a native understanding

Table 1 Study informants

Role	Description of profile	Organization
P1. Chief technology officer	Experience as both a software developer and a technology leader. Involved in leading technology strategy and vision for scalable modern software platforms.	Nation-wide IT-focused business
P2. Business lead	Responsible for overseeing the development, testing, and production of embedded products (shipped in millions) and has helmed full stack (including hardware) software product development in multiple countries.	International technology company
P3. Software expert	20 + years of experience as a software developer in an international IT company and has worked in security- and performance-critical system testing, engineering, and optimization. Also, an expert in automating testing procedures.	International technology company
P4. Software expert	Has worked in several software consulting companies and in many developer roles, including a standard developer and a scrum master. Currently works as a people lead, guiding development teams toward best practices.	An international software consultancy
P5. Technical AI expert	Extensive experience in leading and developing data science and AI related projects. In-depth expert on data science and analytics.	International software consulting company
P6. Technical AI expert	In-depth expertise in ML and analytics and involved in company transition towards AI tools.	International insurance company
P7. Analytics & AI consultant	20 + years of industry experience in analytics and AI and a top consultant for AI companies.	Several startups
P8. Software architect	Responsible for the design, development, and operation of both new systems and legacy products within the company.	Nation-wide fast food chain
P9. Business area lead	Experience in overseeing and consulting projects, primarily in the healthcare sector. Responsible for increasing companies' maturity level in adopting ML technologies into practice.	International software consulting company
P10. AI system developer/ business consultant	Previously worked as a systems manager, although has been shifting more towards AI system development. Currently employed as an ML system designer and developer.	Global technology consultancy
P11. AI system developer/ team lead	Has been leading teams of data scientists for 10 + years in two prominent software companies.	Global technology consultancy
P12. Cybersecurity expert	Responsible for ensuring the cybersecurity of security-critical systems. Works closely with DevOps pipelines and built-in security within the organization.	Governmental organization
P13. Professor	Responsible for AI education at the university and an expert on the societal impact of AI.	Large university
P14. Professor	Listed among the top 100 IT authorities in his country with extensive industry collaboration over many decades.	Large university
P15. Professor	Listed among the top 100 IT authorities in his country with extensive industry collaboration over many decades.	Large university
P16. Professor	In-depth AI expert and leader of an industry–academia research project on AI in healthcare. Research on intelligent systems and AI for 15 + years and is currently responsible for organizing multidisciplinary AI studies within the university.	Large university
P17. Professor	Has extensively studied software development lifecycles for 20 + years in multiple countries, and has been in a key position to witness trends and trajectories within the field.	Large university
P18. AI researcher	An empirical AI system training algorithm researcher and developer. Research and teaching on AI systems for 15 + years and currently working in a research group developing cutting-edge ML technologies.	Large university

of the research context that helped in interpreting the data. Eventually, 18 experts agreed to be interviewed online for this study. The background information on informants is presented in Table 1, although the organizations are only described on a general level to protect informant anonymity.

The interviews were structured to incorporate two main themes: (1) trends and changes in software development, and (2) the drivers and consequences of these changes. Based on the informant responses, we also asked clarifying questions, if required. The interview protocol is provided in Appendix A. The informants were interviewed through online video calls (lasting between 45 and 90 min) by the first author. The interviews took place in the first quarter of 2021, and all were recorded and subsequently transcribed. In addition to the transcriptions, additional notes were taken during and immediately after the interviews.

3.2 Data analysis

We employed the Gioia method (Gioia et al., 2013) to guide the data analysis. As stated by Gioia and his colleagues (2013), novel insights can often be obtained by carefully examining how different actors experience events. Gioia et al. (2013) further suggested certain practices that bring "qualitative rigor" to the analysis process. Moreover, the Gioia method is a well-established approach for analyzing

Table 2 Key concepts and associated codes with examples

Concept	Examples of codes	Illustrative quote
Low code/no code development environments and shift towards higher abstraction languages	Low code, no code, visual programming tools, Java, JavaScript, functional programming	*"Companies doing the robot process automation (RPA) have largely transitioned into low code/no code"* (P7) *"Out of the development work we do here it's nowadays mostly all JavaScript and systems used with a browser. That's just the fastest and easiest."* (P4)
Automation favoring development and testing	Automation, DevOps, MLOps, automated testing, automation tests	*"Now there is this new Business Devops, BizDevOps, and the idea that everything has to be brought under the same cycle…"* (P15) *"[DevOps] has radically changed a lot of things yes"* (P6) *"Automation tests enable testing much more, but the automation of automation testing is a regression test. We have to test our automation tests. So when we develop these new things we need to still involve human intelligence"* (P3)
Availability of user behavioral data and data on system performance	User profiling, profiling data training data, test data, training data sets	*"For example, online retailers collect a lot of data about customers. What happens in the online shop is of course the baseline but then through cookies and other means they collect a lot of information about what people do online in general."* (P7) *"People from our embedded side provide us with data, which we then use to train models that can produce more useful information from the product"* (P2) *"When using medical data to train models you have to have all agreements in order, which creates a bureaucratic and regulated step [in the ML system development process]"* (P18)
APIs for ML and availability of algorithm resources	Keras, TensorFlow, PyTorch, ML API, R, Python, Open source, ArXiv, open access, open repository, free software,	*"In practice there are these two, R and Python and R is where you can do pretty much all the experimenting, but when you build the product you use python."* (P6) *"I'm not completely sure but I understand Keras and TensorFlow are completely open source and anyone can go through them."* (P16) *"The cloud services also rely heavily on this open source development ecosystem"* (P5)
Digital elements (e.g., sensors) infused in physical products	Sensors, IoT, products, cyber-physical,	*"Our products are more and more bytes than just pieces of metal. And the digital elements are more and more important in terms of customer value." (P7)* *"Looking at sports teams like Bayern München or Red Sox they collect sensor information of every single footstep taken on the playing field etc."* (P9)
Cloud computing services support hardware management	Platform as a service, infrastructure as a service, infrastructure as code,	*"For example, Google Cloud markets that they have TPUs that are particularly good for tensor and matrix calculations, and here is the API for using these, and that way we have quite easy tools for training ML models"* (P4) *"In DevOps, ScienceOps, MLOps, they all follow the same principles, so we build the CI/CD pipeline, then you have infra as a code, quality monitoring and all that."* (P11)
Security-critical and privacy-sensitive systems are being built on cloud platforms	Use of cloud services, sensitive information, security-critical systems	*"For consultancies we often build systems with the tools that our customers have already selected, and they often have existing deals with cloud services that suit their needs"* (P9) *"There are certain legal requirements, so we have to abide by those, but yes, it's the cheapest and easiest to just build systems on cloud [platforms]"* (P12)
Cloud services as expensive and difficult to create	Costs of providing cloud computing services	*"It would be a waste [to not use cloud computing services]. They are big products, widely tested, and not easy to do ourselves."* (P13)
Dominance of a few cloud services	AWS, Google Cloud, Microsoft Azure, Huawei Cloud, Alibaba Cloud, Tencent Cloud	*"As software providers we primarily work with the three services: AWS, Google Cloud, and Azure"* (P11) *"AWS is an open platform, but if you want to take your project from there and host it somewhere else, it won't happen with just a snap of fingers."* (P4)

and reporting qualitative research that has also been adopted in previous IS literature (e.g., Alshahrani et al., 2021; Mäntymäki et al., 2020). Typical of inductive research, the analytical process was iterative and partially overlapped with the data collection. Nevertheless, certain phases of the analytical process could be recognized, during which we iterated and refined inferences of theoretical mechanisms from the empirical material.

We started with open coding (Strauss & Corbin, 1998), and the first stage of the analysis process included reading the interview transcripts and assigning codes to describe the content of the interviews. We searched for all instances where technology trends were discussed, identifying unique trends and related phenomena. Beyond coding, we identified differences and similarities among segments in the empirical data, as indicated by the thematic format of the interviews. This practice was similar to constant comparisons in grounded theory research (Strauss & Corbin, 1998). The outcome of this step was the identification of 14 1st order concepts of technology trends. These concepts, associated keywords, and example quotes are displayed in Table 2.

In the second stage of the analysis process, we classified any technology trends identified during the first round of coding into broader concepts, while making notes throughout the process to document the choices made and further develop our insights. Typical of an iterative research process, we refined our coding procedures according to our evolving understanding (Strauss & Corbin, 1998). In the third stage of the analysis process, we incorporated the nonmaterial culture dimension from cultural lag theory (Ogburn, 1957) into the analysis and particularly focused on how the influence of material culture (technology trends) on nonmaterial culture manifests in the categories presented in the second stage of the process. This resulted in the emergence of four theory-guided aggregate themes that represent technology-driven themes in the evolution of nonmaterial culture. These themes will be elaborated on in the next section, while the three stages of the analysis process are summarized in the data structure (Gioia et al. 2013) presented in Fig. 1.

4 Findings

Four theory-guided aggregate dimensions emerged as a result of the analysis (see Fig. 1): (1) shift from manual tasks to scalable solutions, (2) increased emphasis on data, (3) convergence of IT and non-IT industries, and (4) cloud as the dominant computing paradigm. These themes are described in the following subsections. First, we elaborate on the technology trends, and then we connect them to the identified aggregate dimensions representing change in nonmaterial culture.

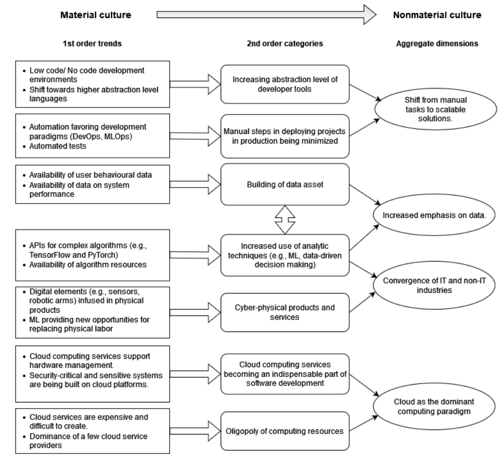

Fig. 1 The results of the data analysis

4.1 Shift from manual tasks to scalable solutions

The informants noted that the abstraction level of development tools across domains within the software industry has constantly increased, and this trend can be expected to continue. This means that while there is (and will be) a need for developers throughout the software stack, from the operating system kernel to the highest abstraction-level user space applications, the proportion of development work that takes place on the high levels of the stack increases. One of the developments that may boost this is the proliferation, advancement, and broad application of ML (deep learning in particular). For example, this was noted by informant P5, who stated the following regarding the future of software development:

> *"There is this idea of Software Development 2.0 and connected to languages like Swift, and the idea here is that you use ML to approximate any function. Because ML is essentially just approximating some function."* (P5).

The growing abstraction level is also visible in the popularity of programming languages, where there has been a shift from low-level languages (e.g., C/ C++) to virtual machine-based languages (e.g., Java), and further towards scripting languages primarily intended for web environments (e.g., JavaScript). The informants also mentioned and further speculated about the role of visual programming tools, which appear to be fundamentally present in various development platforms. On this topic, informant P6 provided the following explanation:

> *"We use SAS enterprise guide that has a graphical user interface, and we just press buttons and it does the SQL queries and all that automatically. This helps*

> *people who have no prior experience in SAS to be able to contribute faster than with old code-based SAS-versions."* (P6).

In addition to programming languages and tools (such as SAS), the increasing abstraction level in development tools can be seen in the emergence of various development platforms that manage a multitude of aspects for developers. The informants discussed low-code and no-code environments in robotics (e.g., RPA), game development (e.g., Unity, Unreal), and web development (e.g., Drupal and WordPress) as examples of how the abstraction level in developer tools has increased. As an example, informant P7 stated the following:

> *"Companies doing robot process automation (RPA) have largely transitioned into low code/no code. So there, the skill requirement for getting certain things done is lower. I actually see the development as a camel with two humps. On the first is the developer tool developers, who are highly skilled and specialized, and then on the other we will have low code/no code developers"* (P7).

In addition to developer tools, the proliferation of the DevOps/MLOps paradigm and related technologies is another major technological driver of the reduction of manual labor within software development, as indicated by the informants. For example, popular online repositories (such as GitHub, GitLab, and BitBucket) provide support for using DevOps with CI/CD pipelines, cloud service providers have online lectures on how to build MLOps pipelines on their services, and tools such as Docker have become a standard within most software development projects. Informant P3 stated the following:

> *"There are essentially… or I mean in essence, we have two steps in virtualization. First, there was the shift from physical servers to cloud, and this has already happened. Then there is now the container world that you run all software in containers, and this will likely stay to some degree, but I don't know if containers are suitable for all corners of larger software systems"* (P3).

The informants also discussed the occurrence of shifts within project management to accommodate DevOps and enable it to be employed efficiently. In this transition, software developers seem more eager to start using DevOps in full, whereas customers appear to be more skeptical regarding the potential benefits of DevOps, according to our informants. This issue is articulated in the following quotes:

> *"We are fully using DevOps, but some of our customers are skeptical and do not understand why we need to update [our product] constantly. [They ask] can't we just make a solid product and that's that?"* (P2).
> *"Even if software is developed in fast cycles, the customers may not appreciate a new update every day. Not even every month, and not even every year."* (P17).

One final concept related to this theme was that of microservices and the idea of utilizing premade components for building complex systems rapidly. This trend was fueled by the availability of free software blocks, the architectural trend of creating "mosaic software," and the increasing abstraction level of development work. Informant P4 discussed this process as follows:

> *"When Unix command line tools do that one thing well, you can chain the commands together or use the outcome of one command in the next command. Similarly, in microservices, if the responsibility limits are well set, then in the best cases you can build bigger working systems by using smaller blocks."* (P4).

Overall, these technology trends were driven by (and connected to) the nonmaterial cultural trend of a shift from manual tasks to scalable solutions. The informants argued that while the increased emphasis on scalability and automation has fueled and directed the formation of these technologies and practices simultaneously, the technologies fuel automation and emphasize scalability in software business. Regardless of the drivers of scalability, the informants perceived that software development as a whole was transforming in such a way that an increasing amount of manual labor was being replaced with automated systems. The main barriers to this change were currently seen to be nonmaterial cultural aspects, such as company culture and developer skills, which connect all the way to IT education. Moreover, this shift toward scalable and automated solutions has implications for developer roles, with increased development time being spent in writing automated tests and making use of available tools and components as much as possible. However, there are also limits imposed by technology and nonmaterial culture pertaining to what can be automated as illustrated by the following quote:

> *"There is unavoidably a limit in what you can fully automate, even in our case, and we are not at the limit yet, but what we are doing is trying to use existing technologies and AI (--) to automate our product as far as reliably possible."* (P1).

4.2 Increased emphasis on data

According to our informants, there is an on-going process in which a once very specialized form of software, ML, is becoming mundane. They argued that creating specific ML systems, such as machine vision tools, no longer requires the expertise of a data scientist, as these systems can be built through relying on pre-built application programming interfaces (APIs). Overall, the informants discussed the following reasons to explain this proliferation of ML: (1) solutions that have made the handling of data and models easier, (2) the availability of processing power, and (3) the use of existing APIs for building ML systems. However, all agreed that ML technology was not even close to its peak, and that these technologies still had significant momentum in academia, industry, and public debate. Many of the contemporary solutions built to support the development of ML systems remove two essential barriers for training ML models: (1) the high technical skill requirement associated with understanding the mathematics behind the training routines, and (2) having access to sufficiently powerful hardware for executing the required computations. Moreover, the proliferation of ML techniques has been rapid, as illustrated by the following quote:

> *"10 years ago, when ML was largely an academic field and we studied random forests and support-vector machines, nobody was interested. Then, suddenly, deep learning became prominent and immediately a narrative surfaced that the age of man is over and Skynet is coming"* (P18).

Despite ML tools and systems becoming more common, the models themselves have increased in complexity. Solutions are also developed by both the industry and the academia for explaining inscrutable ML models, as explained by informant P9.

> *"There's LIME, made by a guy called Marco Tulio Ribeiro and that's used [for explaining ML models]"* (P9).

The informants highlighted that (most) ML algorithms are being published as open source and shared openly via academic repositories (e.g., ArXiv), code repositories (e.g., GitLab, GitHub, and BitBucket), and software forums (e.g., StackOverflow). Premade APIs and frameworks bring complex algorithms to the disposal of programmers with relative ease. For example, the PyTorch and TensorFlow APIs were mentioned by several informants as highly important for the software business in general, as they enable non-specialized software engineers to implement ML systems. Similarly, the role of monitoring tools, tools for data versioning (e.g., DVC and Delta Lake), and various other open source pre-made components have become a standard in software development in recent years. The informants also mentioned an ongoing convergence process between DevOps tools and ML development, discussed as MLOps, where tools such as Azure Machine Learning, Amazon SageMaker, and ML Flow Databricks are used to automatically track every trained model version and the parameters and data employed.

As software engineers globally have access to mostly the same tools for making use of data, high-quality data is becoming an asset that provides companies with a competitive advantage. This means that companies are placing greater emphasis on data collection and curation. Consequently, this could lead to data collection practices that are harmful to consumers, although counter developments have already emerged. For example, legislation such as GDPR has been introduced to protect consumers from rampant data collection practices and any subsequent negative outcomes of data collection, such as privacy violations and personalized cyberattacks that build on information leaks and personal information. Data assets that have been accumulated for years also reinforce the position of leading players, increasing the costs of entering a market for new businesses. On this topic, informant P6 explained the following:

> *"Our company has our own data, and then there is publicly available data. If you are a new company entering the market, then you only have the public data, and that puts you at a disadvantage."* (P6).

While the changes in data-driven development could lead to more data scientists being hired, some informants disagreed. Instead, they felt that the skills of data scientists would simply become part of the skill toolbox of all developers. Furthermore, the experts suggested that the boundaries of specialized developer roles are becoming looser and that individual developers may need to step outside clearly defined boxes (i.e., "UI designer" or "data scientist") to support development work more effectively. Informant P11 explained this topic as follows:

> *"At least all developers should work closely together. (...) Too clearly defined roles in a development team lead to problems sooner or later. Of course, sharing [responsibility] is not always easy either. (...) At some point there might be a situation where you need to call a friend if you're doing something where your own expertise is insufficient."* (P11).

The informants also discussed how ML and deep learning are utilized in increasingly many solutions and systems. Drawing from cultural lag theory (Ogburn, 1957; Ogburn, 1966), despite continual advances in ML and deep learning technologies, they can be viewed as an existing technology that is now being adopted into practice. As the tools and platforms for creating ML systems become readily available, the adoption of ML has shifted focus toward the acquisition and curation of training data. This increased emphasis on data has had various implications for the field of software development. These include the need to recruit personnel responsible for curating data, increasing the maturity level with regards to data collection, resolving legal issues related to data storing, and validating and ensuring the quality of ML system training data. Altogether, the informants suggested that there has been a holistic shift in the nonmaterial culture of software development toward more data-intensive development practices.

4.3 Convergence of non-IT and IT industries

The convergence of non-IT and IT industries was discussed during the interviews in a variety of ways. A recurring theme was the blurring of boundaries between digital and physical products and services toward cyber-physical and increasingly systemic offerings. The following quote by informant P7 illustrates this trajectory:

> *"Digital and physical components (in products and services) are being intertwined, often indistinguishable from one another…so in various so-called 'traditional' industries the offering…I mean the product or service or a combination of them…is in its essence cyber-physical."* (P7).

This convergence (of non-IT and IT industries) was also connected to automation in the way that IT is currently being applied in previously non-IT industries to automate labor that was previously manual. Hence, the informants differentiated between automation in software development (theme 1) and automating manual labor, with the latter being related to non-IT businesses becoming more digital. The participants provided examples, such as machine vision and anomaly detection, which are increasingly used to solve various business problems in non-IT fields. Informant P2 discussed how this process creates competition between novel IT startups and incumbent companies in traditionally non-IT fields:

> *"There is competition between incumbent companies (…) new startups are continuously looking to hog a share of the market. Sometimes, if incumbent companies are slow to adapt to new possibilities, clinging to their old ways, the new companies who have made scalable models from the get go win over, quickly outpacing the incumbents"* (P2).

Cultural lag arose in the interviews when discussing disruptive technologies and the use of IT in traditional industries. According to the informants, companies need to constantly observe and follow IT developments as the advancement of technology is rapid. Moreover, several small, rapid, consecutive improvements to systems can quickly amount to bigger leaps. Informant P18 explained this topic on ML system development as follows:

> *"It's funny to look at how in 2018 (…) we used convolutional networks for biotext mining and got quite good results, and it was quite timely then. But at that time, these new long-short term memory networks became prominent. They were the hottest thing for about a year, but then came these attention models like BERT. So, the advancement cycle of these technologies is really fast."* (P18).

The increasing role of digital elements in various products and services is considered one of the clearest signs of the convergence of IT and non-IT industries. In turn, it is viewed that this increases the demand for developers and other IT staff and constitutes a change in the role of IT functions in organizations. For example, the informants mentioned that several incumbent retail companies are now aggressively hiring developers as more of their business moves online. Similarly, as a result of banks opening more online services, they have less need for customer service personnel and a constantly growing need for IT staff. Moreover, logistics companies are hiring data scientists and investing in IT companies to stay on the edge of self-driving vehicle development. The following quotes from the interviews reflect these changes:

> *"I think they [enterprises who increasingly use and offer IT products] really should employ their own IT people, but when we look at the company landscape today, we do not see this happening in practice"* (P4).
> *"If you look at our products…and the same applies to our direct competitors but also to the whole ecosystem… the digital things become more and more important. (…) Various tools that help operate the products better and more efficiently are being infused to the products themselves (…) and making the digital play together with the non-digital becomes what the customers expect."* (P7).

> *"Previously IT has been some kind of a support service, but today when we look at, for example banks, software is in fact their core service. In this case, it is almost impossible to outsource the programming."* (P8).

While this trend of non-IT businesses transforming into software businesses was pertinent, the interview data indicated that there is a great deal of nuance and complexity involved. First, the trend may not apply to all industry sectors. For example, according to the informants, most service professions are unlikely to be replaced by robots. Second, there is a countertrend emanating from the direction of software consulting businesses, where they wish to sell complete solutions to customers and obtain larger profit margins, instead of renting workers. As companies have an increasing number of IT systems as part of their portfolio, it may be feasible to outsource some development work. Informant P9 gave the perspective of a software consultant company, arguing that it is in their business interest to provide software as a service (SaaS) to customers instead of lending workers:

> *"[Our company] wants to move towards providing entire software and platform products as a service. (...) But for this, we would need to increase the level of our competence to extend beyond mere programming, more towards business transformation and life cycle support."* (P9).

Building on cultural lag theory, the opportunities provided by technology to automate business operations increase pressure on nonmaterial culture to automate manual tasks. However, there are resisting forces, such as the level of maturity within companies to automate tasks and the needs and demands of the workforce. Moreover, the invasion of IT into non-IT industries has created room for various startups to challenge incumbent companies, as fairly stable industry sectors have suddenly been dragged under the influence of rapidly advancing IT systems. The informants provided various examples of brick-and-mortar retailers (e.g., H&M) being challenged by new competitors who have scalable business models designed for the web from their inception (e.g., Zalando and ASOS). These examples suggest that the nonmaterial culture of a company can influence the pace of digitalization within a company by hindering a company's ability to make optimal use of the latest technological affordances.

4.4 Cloud as the dominant computing paradigm

Cloud services and their role in software development were mentioned by almost all informants (often spontaneously and in conjunction with other topics) on multiple occasions during the interviews. Cloud platforms provide a wide range of benefits for developers, ranging from reducing development costs to guiding developers to use well-tested and efficient development practices. The informants maintained that knowledge related to leading cloud platforms has become essential for software engineers. Furthermore, they suggested that other stakeholders, such as company leadership and potential clients of the software or software projects, should also have a general level of knowledge about them. The informants almost unanimously considered cloud services to be an essential part of the software stack of most development projects, and that it was no longer an option for most businesses to not utilize them. Informant P1 elaborated on this as follows:

> *"Various services and frameworks are like a stack, where you have the hardware and infrastructure at the bottom, and in principle the uppermost layers are ready SaaS applications. And the higher up in the stack you operate, the more stuff you have there at the bottom that is made for you. (...) Of course, there is some cost in changing your stack to another, but the alternative of building everything from scratch costs too much. Being free from vendor locks is no longer financially feasible. "(*P1).

These thoughts were echoed by the other informants. Another key trend that was discussed related to the growing role of cloud services in providing guidance to the software development process. Although cloud services initially and predominantly handled the hardware side and nothing else (see e.g., infrastructure as a service), they are currently managing an increasing proportion of the entire software stack. In other words, cloud services are already at the level of providing a platform and software as a service, but their role in the software development business is only expected to increase. For example, informant P11 stated the following:

> *"The cloud services provide premade tools that enable the building of alarms and monitoring [into the software], and we of course use and rely on them heavily."* (P11).

Consequently, knowledge of cloud services has become an important skill to teach at universities as part of software engineering curricula, and a requirement in several job openings in the field of software. The informants also raised concerns that the proliferation of a few cloud platforms (such as AWS, Google Cloud, and Microsoft Azure) has contributed to the materialization of an oligopolistic situation where only a few dominant platforms remain, and where it is difficult for new alternatives to enter the market. Although this trajectory was viewed as somewhat problematic, the informants underscored the importance of the platforms. For example, the dominant role of cloud computing platforms was described as follows:

> *"It would be a waste [to not utilize the big cloud platforms]. They are big products, widely tested, and not easy to do ourselves. (...) I pay for electricity as well, don't I?"* (P13).

Edge and fog computing approaches were perceived as a potential counter trend to the proliferation of cloud computing. The informants viewed privacy and security as the major drivers of these approaches, in addition to being less prone to issues arising from poor or a complete lack of internet connectivity. While there were drivers toward (and away from) cloud computing, the informants were skeptical about a future where the overwhelming majority of computation was not carried in the cloud. For example, informant P5 stated the following:

> *"For quite some time we've discussed edge computing and that edge computing is coming, but so far that trend has not become reality (...) Instead, we seem to continue to move towards cloud computing."* (P5).

Looking at the trend of cloud computing as the dominant paradigm from the cultural lag perspective (Ogburn, 1957), we have already seen clear evidence of businesses reacting to this trend by adjusting their nonmaterial culture. For example, there are observable shifts in the hiring and development practices of software consulting businesses, where increasing emphasis is given to experience with prominent contemporary cloud vendors. Furthermore, a few informants presented evidence regarding the convergence of software development and the development culture promoted by major cloud service providers. More precisely, and as already mentioned, cloud services are taking increased responsibility for how software is made and are producing many instructional videos and offering guidance and documentation, allowing users to make the best use of their systems. Such developments can be seen to further bolster the role of cloud services in the software industry.

5 Discussion

5.1 Key findings

We interviewed 18 experts working in the field of software to elucidate pertinent technology trends. Further, using cultural lag theory, we scrutinize their implications for the nonmaterial culture in the software industry. In our qualitative analysis, we arrived at four aggregate dimensions that can be characterized as technology trends in the nonmaterial culture connected to the software industry, which are summarize in Table 3.

These four aggregate dimensions and their technological drivers have implications for the workforce in the software industry. However, due to the convergence between the IT and non-IT industries, the implications for business are more holistic.

Table 3 Summary of the main findings and their implications for the future of employment

Discovered themes	Description of finding
Shift from manual tasks to scalable solutions	Software development practices and technologies guide development towards automating manual tasks. Technologies such as ML and development practices such as DevOps are crucial to this process. This change is driven by the industry requirement to create scalable, robust, and effective systems. With increased opportunities provided by IT and increased emphasis on scalability across industries, companies that manage to create and operate scalable businesses thrive and outpace the competition.
Increased emphasis on data	The field of ML system development is advancing rapidly and latest advances are being shared openly. Simultaneously, there are advances in hardware to train ML models, as well as the ability to collect, store, and utilize data. Together, these trends contribute to a growing emphasis on data-intensive development practices.
Convergence of non-IT and IT industries	As more industries embrace DT, and as new IT solutions are innovated and become available, a wide variety of manual tasks in traditionally non-software intensive industries are automated. Moreover, IT is becoming an increasingly major part of a wide variety of industries, from logistics to manufacturing. For example, this trend is evidenced by food retailers and airlines transforming into IT companies.
The cloud as the dominant computing paradigm	Despite countertrends such as edge computing, software development has undergone by large a paradigm shift towards web-based systems that are built on top of cloud computing platforms. In conjunction with this transformation, a few cloud platforms have become prominent and achieved what can be characterized as an oligopoly in the market.

5.2 Theoretical implications

Our work offers three key contributions to the IS literature. First, we identified and elaborated on a set of technology trends in the field of software development. While prior literature has focused on identifying trends in software engineering based on factors such as published research articles (Wong et al., 2021) and job advertisements (Gurcan & Cagiltay, 2019; Gurcan & Kose, 2017), we identified trends by analyzing the viewpoints of software professionals from both academia and industry. Our findings confirm the findings of prior studies by demonstrating the importance of skills related to areas such as automation, machine learning, and cloud services (Hemon-Hildgen et al., 2020; Maruping & Matook, 2020; Waizenegger et al., 2020; Wong et al., 2021).

Second, using cultural lag theory, we elucidate how technology trends drive changes in software development and the software industry. With this approach, we demonstrate the feasibility of applying cultural lag theory to understand the implications of pertinent contemporary technology trends on the software industry. This contributes broadly to the IS literature on how technology drives changes in companies and industries (e.g., AL-Zahrani & Fakieh, 2020; Guşeilă et al., 2019; Jääskeläinen et al., 2021; Klünder et al., 2019; Vial, 2019). As an example, with regard to ML and deep learning, our findings support and further expand upon previous work that has described the transformative potential of ML and deep learning (e.g., Brock & Von Wangenheim, 2019; Collins et al., 2021; Dwivedi et al., 2019). This is achieved by providing the perspective of industry practitioners and academics on the transformative and disruptive potential of automating manual labor and transforming development work to be more data-intensive. This is interesting from the viewpoint of the IS literature on AI systems, where automation is the most prominent value of AI. However, the changes ML technologies impose on the culture of software development have received little to no attention (Collins et al., 2021).

Third, our findings advance the understanding of the changing nature of work in the software industry. They provide a comprehensive perspective on different factors affecting how software development is being undertaken in practice, including development practices in DevOps, such as CI/CD (Hemon-Hildgen et al., 2020; Nogueira et al., 2018; Zhao et al., 2017) and technologies that indirectly shape and form development practices, such as cloud computing services (Al-Ruithe et al., 2018) and open data (Grzenda & Legierski, 2021). We argue that this broader perspective on trends in the software industry offers new insights into the future of work through the identification of nonmaterial cultural trends that shape the circumstances surrounding the work of the developer. Furthermore, while our findings emphasized the role of data in the software development process (cf. Mäntymäki et al., 2020), recent work suggests that companies are also embarking increasingly on data-driven decision-making, which is fueled by the growing availability of data and analytics techniques (Zamani et al., 2021). Hence, understanding how to utilize data and analytics has become increasingly important for software development (e.g., Koskenvoima & Mäntymäki, 2015), which has implications for the skills and competences of developers.

5.3 Implications for practice

Drawing on the aggregate dimensions identified in the empirical analysis, we outline the following implications for the future of work in the software industry. First, we expect automation to increase in both software development practices and the systems that are being developed. By creating scalable software from the start, incumbent companies and startups can mitigate any scalability issues that they would otherwise face (Griva et al., 2021; Jääskeläinen et al., 2021). Moreover, our findings suggest that developers will be operating higher on the software stack and with less manual work. Accordingly, the roles of automated testing and governance are highlighted, and the work of developers will probably increasingly consist of creating and validating automated tests that ensure the system works as intended.

Second, with respect to technology competencies, software developers can be expected to increasingly work with data. To make optimal use of data in ML, engineering skills and domain expertise are required. Furthermore, knowledge of cloud computing systems, various software development tools, and ready-made building blocks is becoming increasingly important. By comparison, the ability to write algorithms is shouldered by a relatively small proportion of highly specialized developers. Simultaneously, as new technologies and solutions are created, what is currently considered novel and mystical will be normalized. Moreover, the process of normalization is accelerated by the increased role of cloud services in providing a platform and tools for developers to work effectively and quickly with (and implement) cutting-edge solutions.

Third, due to technologies such as ML offering new business opportunities in non-IT fields, our findings suggest that a growing proportion of software development work can be expected to take place in industries where IT has been employed minimally until now. Moreover, due to growing competition over a skilled IT workforce and remote working opportunities (e.g., Hafermalz, 2021; Hardill & Green, 2003; Zamani & Pouloudi, 2021; Waizenegger et al., 2020), software professionals need personal branding.

Consequently, this may translate into software developers working freelance more often.

Fourth, in addition to these three practical implications, our work has implications for employment of the workforce currently outside the field of IT. Cloud computing, ML, and other major technology trends in software engineering (Akter et al., 2020) are changing the skills that also non-developers are expected to have. In the near future, in many fields, leaders with insufficient understanding of ML and data cannot perform optimally in their work. Consequently, as businesses such as banks and insurance companies become IT houses, their leadership will have to adjust and acquire relevant IT skills. However, due to the increased role of data and the further application of IT across various industries, IT professionals will also be required to accrue the skills and understanding of the application domain in which they create software. An example here is learning analytics (Dennehy et al., 2021) and data analytics (Zamani et al., 2021), where data scientists apply their technical skills to benefit decision-making.

5.4 Limitations and Future Work

As with all empirical studies, our work has limitations that deserve elaboration. First, we synthesized the knowledge of 18 experts in the field of software development, meaning that the results are connected to the views, opinions, and expertise of the informants. We see two potential ways in which future studies could address this limitation: increase the participant pool for reliability by extending the sampling to cover global business leaders or conduct a delphi-style study (Gallego and Bueno, 2014) and ask the experts for comments on the researchers' initial synthesis of their comments.

Second, despite employing rigorous interview sampling and data analysis approaches in our research, there is potential bias in the qualitative analysis due to the data being rich and the authors having to draw their own interpretations. Hence, it is possible that some alternative viewpoints could exist. For example, recent studies have focused on so-called ABCD technologies (artificial intelligence, blockchain, cloud, and data analytics) (Akter et al., 2020). Our findings departed from this by including some technology trends omitted from the analysis (such as DevOps and Low code/No code) while leaving out technologies such as blockchain. The reason for the omission stems from the informants not mentioning blockchain as a major trend, albeit with a slightly different participant sampling process the results may have differed. Accordingly, in addition to the two steps already suggested, future studies could look further into other methodologies and theoretical approaches to supplement our findings. Furthermore, while we described the transformation of the software industry labor market due to influence from contemporary technology trends, this approach is blind to future disruptive technologies and other unforeseen circumstances. Accordingly, technologies are constantly evolving and future research has to stay alert for novel developments.

6 Conclusions

The aim of the current study was to explore *what are the most prominent contemporary technology trajectories in the software industry, and how are they expected to influence the work in software development?* To achieve this, we identified 14 technology trends pertinent to software development. Building on cultural lag theory, we arrived at four aggregate dimensions that describe the ongoing and upcoming changes to nonmaterial culture related to the software industry. These dimensions were as follows: (1) a shift from manual labor to scalable solutions, (2) increased emphasis on data, (3) convergence of IT and non-IT industries, and (4) the cloud as the dominant computing paradigm. Finally, through an analysis of today's technology trends, we discussed how the future of work in software development might transform in the near future.

7 Electronic supplementary material

Appendix A The Interview protocol

Theme	Opening questions	Exemplar continuation questions
1. Trends and changes in software development	What are the most prominent technology trends currently pertinent to software development?	-To whom does this trend concern? -What technologies are related to this trend?
	How do these trends manifest in practice?	-What business opportunities does the trend provide? -Who are the early adopters of this technology?
	What cultural shifts or changes in employment have occurred in the software industry over recent years?	-Whom does this change concern primarily? -Can you give more examples as to where this trend takes place?

Appendix A The Interview protocol

Theme	Opening questions	Exemplar continuation questions
2. Drivers and consequences of the trends	What would you say are the key drivers of [a mentioned megatrend]?	-What other technologies are connected to this? -What other phenomena/trends does this change fuel?
	What implications do [a mentioned megatrend] have on the work of software engineers?	-Can you give an estimate of a time schedule as to when this change is going to take place? -How do you see the situation in 10 years?
	What implications does [a mentioned trend] have on the industry as a whole?	-What business models benefit from this trend? -How do you see developer roles (e.g., data scientist, frontend developer, backend developer, UX designer) in this setting?

Funding Open Access funding provided by University of Turku (UTU) including Turku University Central Hospital.

Declarations

Conflict of interest None.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Akter, S., Michael, K., Uddin, M. R., McCarthy, G., & Rahman, M. (2020). Transforming business using digital innovations: The application of ai, blockchain, cloud and data analytics.Annals of Operations Research,1–33

Al-Ruithe, M., Benkhelifa, E., & Hameed, K. (2018). Key issues for embracing the cloud computing to adopt a digital transformation: A study of Saudi public sector. *Procedia computer science*, 130, 1037–1043

AL-Zahrani, S., & Fakieh, B. (2020). How devops practices support digital transformation. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(3), 1–9

Alshahrani, A., Dennehy, D., & Mäntymäki, M. (2021). An attention-based view of AI assimilation in public sector organizations: The case of Saudi Arabia.Government Information Quarterly,101617

Bankins, S., Formosa, P., Griep, Y., & Richards, D. (2022). AI Decision Making with Dignity? Contrasting Workers' Justice Perceptions of Human and AI Decision Making in a Human Resource Management Context.Information Systems Frontiers,1–19

Bertani, F., Ponta, L., Raberto, M., Teglio, A., & Cincotti, S. (2021). The complexity of the intangible digital economy: An agent-based model. *Journal of Business Research*, 129, 527–540. https://doi.org/10.1016/j.jbusres.2020.03.041

Bianchi, M., Marzi, G., & Guerini, M. (2020). Agile, Stage-Gate and their combination: Exploring how they relate to performance in software development. *Journal of Business Research*, 110, 538–553

Bimber, B. (1990). Karl Marx and the Three Faces of Technological Determinism. *Social Studies of Science*, 20(2), 333–351. https://doi.org/10.1177/030631290020002006

Brinkman, R. L., & Brinkman, J. E. (1997). Cultural lag: Conception and theory. *International Journal of Social Economics*, 24 No(6), 609–627

Brinkman, R. L., & Brinkman, J. E. (2006). Cultural lag: in the tradition of Veblenian economics. *Journal of Economic Issues*, 40(4), 1009–1028

Brock, J. K. U., & Von Wangenheim, F. (2019). Demystifying ai: What digital transformation leaders can teach you about realistic artificial intelligence. *California Management Review*, 61(4), 110–134

Collins, C., Dennehy, D., Conboy, K., & Mikalef, P. (2021). Artificial intelligence in information systems research: A systematic literature review and research agenda. *International Journal of Information Management*, 60, 102383

Dafoe, A. (2015). On technological determinism: a typology, scope conditions, and a mechanism. *Science, Technology, & Human Values*, 40(6), 1047–1076

Dennehy, D., Conboy, K., & Babu, J. (2021). Adopting Learning Analytics to Inform Postgraduate Curriculum Design: Recommendations and Research Agenda.Information Systems Frontiers,1–17

Drahokoupil, J., & Fabo, B. (2016). The platform economy and the disruption of the employment relationship.ETUI Research Paper-Policy Brief, *5*

Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. *International Journal of Information Management*, 48, 63–71

Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., et al. (2019). Artificial intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy.International Journal of Information Management, 101994

Estevam, A., Dennehy, D., & Conboy, K. (2020). Using Flow Tools to Enact Control in Software Development Projects: A Cross-case Analysis. *Information Systems Frontiers*. https://doi.org/10.1007/s10796-020-10081-w

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological forecasting and social change*, 114, 254–280

Frick, N. R., Mirbabaie, M., Stieglitz, S., & Salomon, J. (2021). Maneuvering through the stormy seas of digital transformation: the impact of empowering leadership on the ai readiness of enterprises.Journal of Decision Systems,1–24

Gallego, D., & Bueno, S. (2014). Exploring the application of the delphi method as a forecasting tool in information systems and technologies research. *Technology Analysis & Strategic Management*, 26(9), 987–999

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organizational research methods*, 16(1), 15–31

Griva, A., Kotsopoulos, D., Karagiannaki, A., & Zamani, E. D. (2021). What do growing early-stage digital start-ups look like?

A mixed-methods approach.International Journal of Information Management,102427

Gubrium, J. F., & Holstein, J. A. (2001). *Handbook of interview research: Context and method*. Thousand Oaks, CA: Sage Publications

Gurcan, F., & Kose, C. (2017). Analysis of software engineering industry needs and trends: Implications for education. *International Journal of Engineering Education*, 33(4), 1361–1368

Gurcan, F., & Cagiltay, N. E. (2019). Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling. *IEEE access*, 7, 82541–82552

Guşeilă, L. G., Bratu, D. V., & Moraru, S. A. (2019). DevOps transformation for multi-cloud IotT applications. *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, IEEE, 1–6

Grzenda, M., & Legierski, J. (2021). Towards Increased Understanding of Open Data Use for Software Development. *Information Systems Frontiers*, 23, 495–513. https://doi.org/10.1007/s10796-019-09954-6

Hafermalz, E. (2021). Out of the Panopticon and into Exile: Visibility and control in distributed new culture organizations. *Organization Studies*, 42(5), 697–717. https://doi.org/10.1177/0170840620909962

Hardill, I., & Green, A. (2003). Remote working—Altering the spatial contours of work and home in the new economy. *New Technology, Work and Employment*, 18(3), 212–222. https://doi.org/10.1111/1468-005X.00122

Healy, J., Nicholson, D., & Parker, J. (2017). Guest editors' introduction: technological disruption and the future of employment relations

Hemon-Hildgen, A., Rowe, F., & Monnier-Senicourt, L. (2020). Orchestrating automation and sharing in DevOps teams: a revelatory case of job satisfaction factors, risk and work conditions. *European Journal of Information Systems*, 29(5), 474–499

Hess, T., Matt, C., Benlian, A., & Wiesböck, F. (2016). Options for formulating a digital transformation strategy.MIS Quarterly Executive*15*(2)

Islam, A. K. M. N., Mäntymäki, M., & Turunen, M. (2019). Why do blockchains split? An actor-network perspective on Bitcoin splits. *Technological Forecasting and Social Change*, 148, 119743

Jääskeläinen, A., Yanatma, S., & Ritala, P. (2021). How Does an Incumbent News Media Organization Become a Platform? Employing Intra-Firm Synergies to Launch the Platform Business Model in a News Agency, Journalism Studies, DOI: https://doi.org/10.1080/1461670X.2021.1979426

Klünder, J. A. C., Hohl, P., Prenner, N., & Schneider, K. (2019). Transformation towards agile software product line engineering in large companies: A literature review.Journal of Software: Evolution and Process, 31(5), e2168

Koskenvoima, A., & Mäntymäki, M. (2015). Why do small and medium-size freemium game developers use game analytics? *Conference on e-business, e-Services and e-Society* (pp. 326–337). Springer, Cham

Koutsikouri, D., Madsen, S., & Lindström, N. B. (2020). Agile Transformation: How Employees Experience and Cope with Transformative Change. *International Conference on Agile Software Development* (pp. 155–163). Springer, Cham

Laato, S., Vilppu, H., Heimonen, J., Hakkala, A., Björne, J., Farooq, A. … Airola, A. (2020). Propagating ai knowledge across university disciplines-the design of a multidisciplinary ai study module. *2020 IEEE Frontiers in Education Conference (FIE)*, IEEE 1–9

Magistretti, S., Dell'Era, C., & Petruzzelli, A. M. (2019). How intelligent is watson? enabling digital transformation through artificial intelligence. *Business Horizons*, 62(6), 819–829

Mäntymäki, M., Hyrynsalmi, S., & Koskenvoima, A. (2020). How do small and medium-sized game companies use analytics? An attention-based view of game analytics. *Information Systems Frontiers*, 22(5), 1163–1178

Marshall, K. P. (1999). Has technology introduced new ethical problems? *Journal of business ethics*, 19(1), 81–90

Maruping, L. M., & Matook, S. (2020). The evolution of software development orchestration: current state and an agenda for future research. *European Journal of Information Systems*, 29(5), 443–457

Matt, C., Hess, T., & Benlian, A. (2015). Digital transformation strategies. *Business & Information Systems Engineering*, 57(5), 339–343

Meuser, M., & Nagel, U. (2009). The expert interview and changes in knowledge production. *Interviewing experts* (pp. 17–42). London: Palgrave Macmillan

Nogueira, A. F., Ribeiro, J. C., Zenha-Rela, M. A., & Craske, A. (2018, September). Improving la redoute's ci/cd pipeline and devops processes by applying machine learning techniques. *2018 11th international conference on the quality of information and communications technology (QUATIC)* (pp. 282–286). IEEE

Ogburn, W. F. (1957). Cultural lag as theory. *Sociology & Social Research*, 41, 167–174

Ogburn, W. F. (1966). *Social change with respect to cultural and original nature* (No. HM101 O4 1966)

Pappas, I. O., Mikalef, P., Giannakos, M. N., Krogstie, J., & Lekakos, G. (2018). Big data and business analytics ecosystems: paving the way towards digital transformation and sustainable societies. *Information Systems and e-Business Management*, 16(3), 479–491

Reis J., Amorim M., Melão N., Matos P. (2018) Digital Transformation: A Literature Review and Guidelines for Future Research. In: Rocha Á., Adeli H., Reis L.P., Costanzo S. (eds) Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. *Advances in Intelligent Systems and Computing*, vol 745. Springer, Cham. https://doi.org/10.1007/978-3-319-77703-0_41

Schneckenberg, D., Benitez, J., Klos, C., Velamuri, V. K., & Spieth, P. (2021). Value creation and appropriation of software vendors: A digital innovation model for cloud computing. *Information & Management*, 58(4), 103463

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research techniques* (pp. 1–312). Thousand oaks, CA: Sage publications

Suominen, A., Hyrynsalmi, S., & Knuutila, T. (2014). Young mobile users: Radical and individual–Not. *Telematics and Informatics*, 31(2), 266–281

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J. Q., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, 122, 889–901

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, 28(2), 118–144

Waizenegger, L., McKenna, B., Cai, W., & Bendz, T. (2020). An affordance perspective of team collaboration and enforced working from home during COVID-19. *European Journal of Information Systems*, 29(4), 429–442. https://doi.org/10.1080/0960085X.2020.1800417

Wang, H. Y., Wang, J. H., Zhang, J., & Tai, H. W. (2021). The Collaborative Interaction with Pokémon-Go Robot uses Augmented Reality technology for Increasing the Intentions of Patronizing Hospitality.Information Systems Frontiers,1–13

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2020). Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. *European Journal of Information Systems*, 29(5), 458–473

Wong, W. E., Mittas, N., Arvanitou, E. M., & Li, Y. (2021). A bibliometric assessment of software engineering themes, scholars and institutions (2013–2020).Journal of Systems and Software,111029

World Bank (2018). World development report 2019: The changing nature of work. ONLINE, available at: https://elibrary.worldbank.org/doi/abs/10.1596/978-1-4648-1328-3

Wu, S. Y. (2019). Key technology enablers of innovations in the ai and 5 g era. *2019 IEEE International Electron Devices Meeting (IEDM)*, IEEE 36–3

Xi, N., Chen, J., Gama, F., Riar, M., & Hamari, J. (2022). The challenges of entering the metaverse: An experiment on the effect of extended reality on workload.Information Systems Frontiers,1–22

Zamani, E. D., Griva, A., Spanaki, K., O'Reilly, P., & Sammon, D. (2021). *Making sense of Business Analytics in project selection and prioritisation: Insights from the start-up trenches*. Information Technology & People. https://doi.org/10.1108/ITP-09-2020-0633

Zamani, E. D., & Pouloudi, N. (2021). Shared mental models and perceived proximity: A comparative case study. *Information Technology*. https://doi.org/10.1108/ITP-02-2020-0072. & People, ahead-of-print (ahead-of-print)

Zhao, Y., Serebrenik, A., Zhou, Y., Filkov, V., & Vasilescu, B. (2017, October). The impact of continuous integration on other software development practices: a large-scale empirical study. *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 60–71). IEEE

Zhu, X., Ge, S., & Wang, N. (2021). Digital transformation: A systematic literature review. *Computers & Industrial Engineering*, 162, 107774

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Samuli Laato is a Post-doctoral Researcher at Tampere University, Finland. Prior to his career in academia he worked as an educational game programmer. Laato has worked in various multidisciplinary research projects and has experience in a wide variety of research methods. He has published research in, for example, *European Journal of Information Systems, Computers in Human Behavior, Internet Research, and Telematics & Informatics*. Laato has over 60 peer-reviewed publications.

Matti Mäntymäki is an Associate Professor of Information Systems Science (Tenure Track) at University of Turku, Finland. His research interests cover a broad range of psychosocial and business implications of digitalization. He has published in outlets such as *Information Systems Journal, Technological Forecasting & Social Change, International Journal of Information Management, Journal of Business Research, Information Technology & People, Computers in Human Behavior, Journal of Systems & Software, and Communications of the Association for Information Systems*, among others.

A.K.M. Najmul Islam is an Associate Professor (Tenure Track) at LUT University, Finland. He conducts cross-disciplinary research in the area of digitalization and its impact on citizens, organizations, and society. He is a docent of Information Systems at Tampere University. Islam's publication has appeared in top Information Systems outlets such as Journal of Strategic Information Systems, European Journal of Information Systems and Information Systems Journal. He has published in other highly ranked interdisciplinary journals such as IEEE Access, Computers & Education, Technological Forecasting and Social Change, International Journal of Information Management, Information Technology & People, Computers in Human Behavior, Computers in Industry, Internet Research, Communications of the AIS, among others. He is currently serving as a Senior Editor for Information Technology & People journal.

Sami Hyrynsalmi is an Associate Professor (Tenure Track) of Software Engineering at LUT University, Finland. He holds a D.Sc. (Tech.) from University of Turku, Finland. Hyrynsalmi has published in outlets such as *Journal of Systems and Software, Information and Software Technology, and Computers in Human Behavior*, among others.